Role Assignment in Business Process Models

Agnes Koschmider[1], Liu Yingbo[2], and Thomas Schuster[3],

[1] Institute of Applied Informatics and Formal Description Methods
Karlsruhe Institute of Technology, Germany
agnes.koschmider@kit.edu
[2] School of Software, Tsinghua University, Beijing 100084, China
lyb01@mails.tsinghua.edu.cn
[3] FZI Forschungszentrum Informatik, Germany
schuster@fzi.de

Abstract. Business processes are subject to changes due to frequently fluctuating opportunities. The changes has as result a modification of business process models and also the organizational model since both models are jointly linked through the assignment of roles to process activities. A consistent adaptation of both model types (due to changes) still poses challenges. For instance, varying competences and skills are insufficiently considered for the (re-)assignment of roles to process activities. As a consequence, tasks are performed inefficiently. In this paper we will present an organizational model that considers resources' competences, skills and knowledge. Based on this model the hidden Markov model is applied to efficiently assign roles to process activities. The improvement in task processing through automated role assignment is a significant contribution of this approach.

1 Introduction

A business process model consists of activities that are performed by roles or respectively by organizational units. The assignment of roles to process activities depends on the roles' skills and competences and should ensure that information is allocated to proper persons. For instance, a secretary should be assigned to tasks doing preliminary work for seniors. A salesperson should be assigned to tasks supporting the interaction with customers.

Changes in information system requirements or new business opportunities may require modifications of process activities and the assignment of roles to them. Role assignment tends to be complicated because roles might be assigned to hundreds of activities as illustrated by the following example. In the past the following observations were made in enterprises [1, 2], e.g., *Enterprise A* had 48 roles and 922 process activities; in *Enterprise B* 102 roles were allocated to 399 process activities and in *Enterprise C* 81 roles were allocated to 256 activities. Advanced business process model experiences are required in order to understand and rapidly assign appropriate roles to business process activities. Therefore, assisting process modelers to efficiently assign roles to activities is of great value.

F. Daniel et al. (Eds.): BPM 2011 Workshops, Part I, LNBIP 99, pp. 37–49, 2012.

We aim to improve the assignment of roles to process activities through an advanced meta-model for resources that considers roles' competences, skills and knowledge. Based on this model we are capable to efficiently retrieve appropriate roles to perform a task. The retrieval and assignment rely on the hidden Markov model [3]. The advantage of using a hidden Markov model instead of other approaches (like data mining based approaches) is the ability to consider role's competences and workflow history data for role assignment. Additionally, the model allows considering the relationship between different process activities (described via control flow) rather than focusing on a single activity.

The approach presented in this paper can be applied twofold. Assume the process modeler is creating a process model and she is uncertain which role to assign to a process task. Based upon our approach recommendations of appropriate role assignment can be made. On the other hand our approach is suitable to support exhaustive process model reuse. Before creating a new process model by assembling already designed process models, the process builder can use our approach to update the reused process model. As soon as a process builder reuses a process model, the role assignment of the model is matched with our algorithm.

The meta-model for organizational units is summarized in the next section. Section 3 illustrates our approach of role assignment to process activities. The application of our approach is demonstrated in Section 4. Section 5 compares our approach with related work and Section 6 concludes the paper with an outlook on future research.

2 Modeling Foundation

In this section we illustrate the requirements and modeling foundation of our proposed solution. Therefore we will outline a meta-model for the description of resources that can be utilized in business processes. The meta-model defines the resource modeling language (RML), which is introduced by [21]. Within the following subsection we illustrate the core of RML defined by the human resource meta-model (HRMM).

2.1 Organizational Meta-model

HRMM is a MOF-compliant meta-model, modeled as ecore model [20]. An overview of the HRMM is given in Figure 1. Central concepts of HRMM are: *HumanResource*, *Role*, *OrganizationalUnit* and the competence related modeling objects *Competence*, *Skill* and *Knowledge*. In utmost related approaches competence concepts are not modeled explicitly, although different studies revealed that roles and human resources depend on competences [11, 16]. To tackle this issue HRMM integrates competence descriptions and associates them to roles and human resources, thus allowing for enhanced assignment strategies. In HRMM this is represented by the model elements *Competence*, *Skill* and *Knowledge*. In order to enable a sound assignment of activities to resources, we will reveal relationships of competence models and resource models (a business process view) that can be modeled in RML.

HRMM is part of the RML meta-model and combines approaches known by other resource meta-models in business process management [11] with competence descriptions as utilized in human resource management [17, 18]. The meta-model allows definitions of organizational aspects and hierarchies; furthermore it allows an explicit extension of these structures by descriptions of competences (in particular competences, skills and knowledge). The competences can be modeled independently and may be reused for further specifications of roles and human resources (as shown below). With the intention of empowering assignment strategies, it is essential to know that competences, skills and knowledge can be detailed by a level of proficiency (as given in EQF [17]); furthermore competences may require other competences, skills and knowledge, while skills can require other skills and knowledge. Additionally competences can be prioritized by a correlation coefficient.

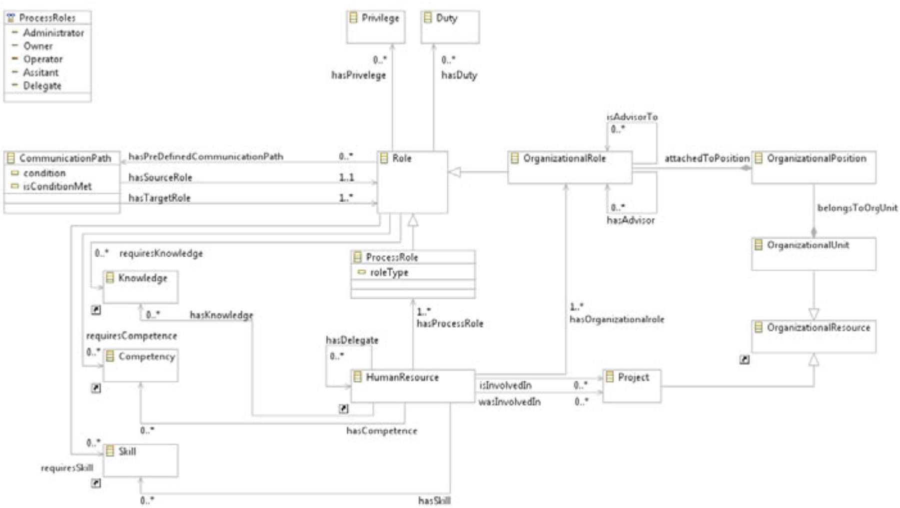

Fig. 1. Human Resource Meta Model (HRMM)

Human resources (*HumanResource*) are used to depict manpower. To represent their integration in organizational structures human resources can be associated to organizational roles (*OrganizationalRole*) and positions (*OrganizationalPosition*). Organizational hierarchies are detailed and reflected by the elements *OrganizationPosition*, *OrganizationalUnit* and their associated relationships *hasAdvisor* and *isAdvisorTo* of *OrganizationalRole*. Roles may be detailed by rights (*Privilege*), obligations (*Duty*) and predefined communication channels (*CommunicationPath*), e.g., to model escalation mechanisms. Furthermore organizational roles can be used to determine appropriate resources for task execution. Figure 2 illustrates an instance of a human resource model given in RML. While it is obvious that the organizational structure is basically given graphically, the box at the right hand side of the figure reveals that a lot of properties are modeled as non-graphical attributes of a model element.

The description of a delegate is given as non-graphical attribute (in this case *Architect B* – with associated organizational role *Chief Designer* – is a delegate of *Manager A* – with associated organizational role *Project Manager*). Furthermore the description of competences, skills and knowledge is also modeled as non-graphical attributes. The competences given in Figure 2 are competences suggested by the European standard e-CF [17].

Obviously the combination of the concepts states above bridges the gap between business process management and human resource management. Consequently, not only details about the modeled resources and their competences are revealed, but also decision support for a multitude of questions is facilitated. Decisions which can be supported by this modeling technique are (1) Recruitment of new resources, (2) Identification of core competences or (3) Task allocation.

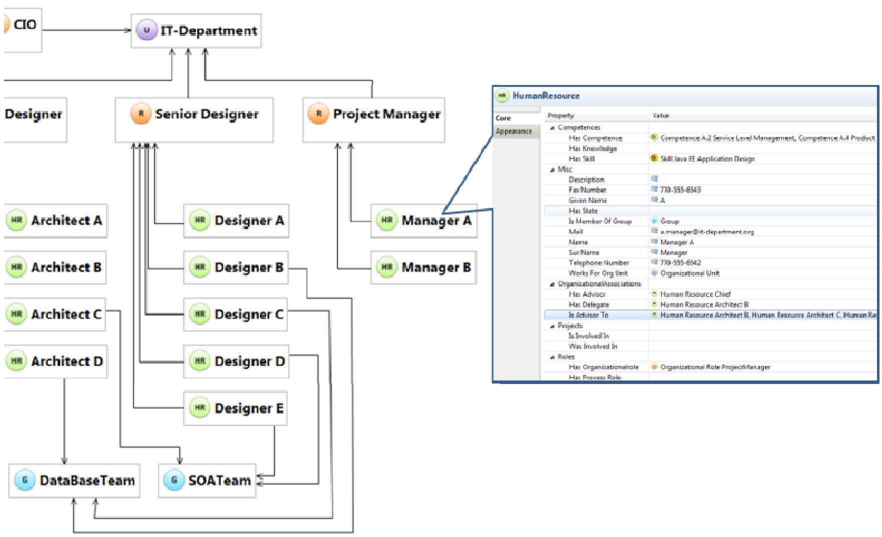

Fig. 2. Cut-Out of a RML Instance

The first two points are strongly related to HR planning and business partner selection. Task allocation finally is a common issue during business process execution – usually solved by the definition of declarative constraints (at modeling time) and specialized algorithms to match appropriate resources (at runtime). In the following sections task allocation is addressed by assignment of roles to process activities instead of particular resources. Thereby the process model is kept independent of changes in the organizational model.

3 Automatic Role Assignment Based on the Hidden Markov Model

In this section we suggest to solve the assignment of roles to activities by means of hidden Markov model inference [4].

3.1 Hidden Markov Model Inference

The hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. These states are not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output events. Therefore the sequence of events generated by the HMM gives some information about the sequence of states. Note that the adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model; even if the model parameters are known exactly, the model is still 'hidden' [3].

Formally, a HMM can be defined by a set of parameters $\lambda = (N, M, T, E, \pi)$:

- N, is a set of hidden states,
- M, is a set of events,
- $T_{|N|\times|N|}$, is a transition matrix that determines the state transition probability,
- $E_{|N|\times|M|}$, is a matrix that denotes the emission probability that the event will be observed for any given state $n \in N$,
- π, is an initial vector that denotes the probability of each state in the first beginning.

For a given sequence of events with *t* observations $y_{1..t} \in M$, and a hidden Markov model with parameter λ, an inference associated to this HMM [4] is to find a probability distribution over hidden states for a point in time in the past, i.e. to compute $P(x_k|y_{1..t})$, for $k < t$. This inference problem can be solved by the so called "forward-backward algorithm", which is an efficient method for computing the smoothest values for all hidden state variables [5].

3.2 Building Hidden Markov Model

In our approach the set of hidden states N describes the set of possible roles, which may be attached to activities in a workflow model. The set of observable events M relates to a set of workflow activities. The inference associated to HMM can be described as the probability distribution over roles (hidden state) and activities of a given activity sequence (observed events) in a process model and a hidden Markov model. By this probability distribution the likelihood of the assignment of roles to activities can be determined. Hence, an ordered list of role assignments can be recommended to the process modeler according to the probability distribution.

As we mentioned before, if parameter λ of HMM is defined the probability distribution over roles and activities can be easily obtained by the "forward-backward algorithm". In conclusion determining the parameter λ has to be done; therefore we present an approach to fulfill this task:

- Let all candidate roles (plus a start and an end role) be the set of hidden states, namely N,
- Let all activities (including a start and an end activity) be the set of events, namely M,

- Transition matrix T can be obtained from event logs of the workflow by analyzing the sequence frequency of role transitions. The entry T_{ij} represents the probability of transition from role i to role j, thus T_{ij} can be calculated as:

$$T_{ij} = \frac{freq(R_i \to R_j)}{freq(R_i \to *)} \quad (1)$$

$freq(R_i \to R_j)$ refers to the number of transitions from *role i* to *role j* in the log file, and $freq(R_i \to *)$ refers to all transitions from *role i*.

- Emission matrix E can be specified according to the competence and skills of roles for a given activity, which allows answering the following question: "according to the knowledge and skill of a *role i* what kind of activities are suitable to be performed by this *role*?". This competence value can be obtained by the measurement of the human resource meta-model (see Section 2).

$$E_{ij} = \textit{Competence of role i to activity j} \quad (2)$$

- Finally, the initial vector π is (1, 0, …, 0); which indicates that the start activity is always performed by the start role.

Before calculating the role assignment probability, we need to uncover similar process activities in order to avoid inconsistencies in the workflow event log. To find synonyms, homonyms and different abstraction levels of activity labels, we use the similarity measures presented in [22]. After this similarity match process, activities in the new workflow model can be easily mapped to events in the hidden Markov model.

3.3 Calculating Role Assignment Probability Matrix

After obtaining the parameters of the HMM and uncovering similar process activities, the final step is to determine the probability of roles being appropriate to be assigned to activities. However, in real world a workflow model usually contains various control flow structures such as e.g., *joins* and *forks*, which eventually result in multiple observed sequences of activities in the same workflow model. Such observed sequences may generate multiple probability distributions over roles for activities, when HMM inference is applied. Therefore, it is essential merging different probability matrixes for each observed activity sequences.

Assume that the workflow model is a directed acyclic graph (DAG) of activities. For each activity $a_i \in M$ let probability distribution p_i be a vector of probability with ||N|| entries. Each entry in p_i refers to the probability of one role for activity a_i. Based on a workflow model, an activity sequences set S can be generated by enumerating all paths from the start activity to the end activity.

For each activity sequence $s \in S$, a probability matrix P_s of roles over activities can be computed by means of the "forward-backward algorithm" namely $P_s = (p_1, p_2, \ldots p_n)$, n=||M||. Note if activity a_j is not in the given sequence s, then the corresponding probability distribution vector p_j=0 is $P_s = (p_1, p_2 \ldots p_{j-1}, 0, p_{j+1} \ldots p_n)$. In addition, if the occurrence probability of an activity sequence is different, a weight w_s can also be assigned to the activity sequence s. Once the probability matrix for any activity sequence has been computed, the probability distribution matrix P of roles over activities in the workflow model can be obtained by calculating the weighted average probability matrix of all the probability matrixes and normalizing each column vector in order to ensure that each column sum equals 1.

$$P = normalize\left(\frac{\sum_{s\in S} w_s P_s}{\sum_{s\in S} w_s}\right) \quad (3)$$

Based upon the role assignment probability matrix, the role assignment for any given activity can be easily performed by retrieving the most appropriate role for a specific activity.

4 Assignment Demonstration

In order to demonstrate the applicability of our approach, we present an example in this section. Let the following process model with seven activities (design activity, verify activity, review prototype, approve design, classify documents and additionally a start and end activity as postulated in Section 3.1.) be given as depicted in Figure 3.

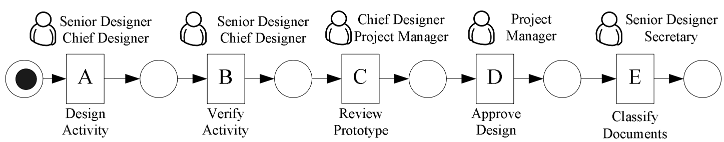

Fig. 3. Example process of engineering design process

There are six roles that can be assigned to these activities {Start Role, Senior Designer, Chief Designer, Project Manager, Secretary, End Role}. Skills and competences of roles are listed in Table 1. Note that the activities {Start, End} and roles {Start Role and End Role} are added in order to facilitate following analysis.

Table 1. Role Information

#	*Role*	*Skill*	*Competence*
R1	Start Role	Start workflow	
R2	Senior Designer	create drawing, classify documents	service
R3	Chief Designer	create drawing, review drawing	design architecture
R4	Project Manager	review drawing, approve design	product planning
R5	Secretary	classify documents	contract management
R6	End Role	End Workflow	

Furthermore we assume that there are ten completed cases in the event log, the case information and related performers' role for each activity are listed in Table 2.

In order to build the hidden Markov model, we perform the following steps. Firstly, let the set of hidden states be the set of candidate roles namely N = {start role(R1), senior designer(R2), chief designer(R3), project manager(R4), secretary(R5), end role(R6)}. Secondly, the event set M is built by observable activities in the process model namely M = {start, design activity, verify activity, review prototype, approve design, classify document, end}.

The transition matrix can also be obtained from the workflow event log by counting the frequency of direct role transition during execution. For example in Table 2 there are 10 direct role transitions from R1 to others (the entries with underscore), 7 out of these 10 transitions are from R1 to R2, and 3 out of these 10 transitions are from R1 to R3. Therefore the entries $T_{1,2}$ and $T_{1,3}$ in transition matrix T are $T_{1,2}$=7/10 and $T_{1,3}$=3/10. Table 3 illustrates the transition matrix calculated out of the workflow log shown in Table 2.

Table 2. Workflow Event Log

	Start	*Design Activity*	*Verify Activity*	*Review Prototype*	*Approve Design*	*Classify Documents*	*End*
1	<u>R1</u>	<u>R2</u>	R2	R3	R4	R5	R6
2	<u>R1</u>	<u>R2</u>	R3	R3	R4	R5	R6
3	<u>R1</u>	<u>R2</u>	R2	R3	R4	R5	R6
4	<u>R1</u>	<u>R2</u>	R2	R3	R4	R5	R6
5	<u>R1</u>	<u>R2</u>	R2	R4	R4	R3	R6
6	<u>R1</u>	<u>R2</u>	R2	R3	R4	R5	R6
7	<u>R1</u>	<u>R2</u>	R2	R3	R4	R5	R6
8	<u>R1</u>	<u>R3</u>	R2	R3	R4	R5	R6
9	<u>R1</u>	<u>R3</u>	R3	R3	R4	R2	R6
10	<u>R1</u>	<u>R3</u>	R3	R4	R4	R5	R6

Table 3. Transition Matrix from Workflow Event Log

	R1:Start Role	*R2:Senior Designer*	*R3:Chief Designer*	*R4:Project Manager*	*R5:Secretary*	*R6:End Role*
R1:Start Role	0	7/10	3/10	0	0	0
R2:Senior Designer	0	6/15	7/15	1/15	1/15	0
R3:Chief Designer	0	1/15	4/15	9/15	1/15	0
R4:Project Manager	0	1/12	1/12	2/12	8/12	0
R5:Secretary	0	0	0	0	0	1
R6:End Role	0	0	0	0	0	1

The emission matrix shows probabilities of generating observable events when the system is in a hidden state. In workflow staff assignment observable events are workflow activities, hidden states are roles. Hence event emission probability means the likelihood of roles to complete certain activities. Apparently, this likelihood is determined by role's skills and competences, therefore the emission matrix can be created by domain experts. Table 4 shows an example emission matrix where each column represents an activity (observable event) and each row represents a role (hidden state). As shown in Table 4, it is most likely for senior designers to complete the "Design Activity" (0.7), while it is quite unlikely to perform the activity of "Approve Design" (0.01).

The initial state vector defines the probability of choosing the first state when the transition starts. Since workflows always start with the start activity the initial role is always the start role; hence the initial state vector is (1, 0…). Once parameters of HMM are defined, role assignment can be easily performed as follows. Assume the process designer tends to reuse process artifacts without assigned roles (in general: appropriate role have to be assigned to modeled process activities). Then all sequences of activities (start to end) are enumerated. Initially, similar activities are uncovered (for instance the activity "Verify Specification" is matched to "Verify Activity", see Figure 4). Subsequently, resulting activity sequences are s_1={Start, Design Activity, Verify Activity, Approve Design, Classify Documents, End} and s_2={Start, Design Activity, Review Prototype, Approve Design, Classify Documents, End}.

Table 4. Emission Matrix for Roles to Activities

	start	*Design Activity*	*Verify Activity*	*Review Prototype*	*Approve Design*	*Classify Documents*	*end*
R1:Start Role	1	0	0	0	0	0	0
R2:Senior Designer	0	0.7	0.2	0.05	0.01	0.04	0
R3:Chief Designer	0	0.2	0.2	0.4	0.15	0.05	0
R4:Project Manager	0	0.05	0.05	0.2	0.65	0.05	0
R5:Secretary	0	0	0	0	0	1	0
R6:End Role	0	0	0	0	0	0	1

With the parameter of HMM defined in the previous discussion, the probability distribution matrix for observed sequence can be computed by means of the "forward/backward algorithm". Table 5 and Table 6 show the probability distribution matrix of s_1 and s_2. Note that for s_1 the review activity is not available, therefore corresponding probability distribution over roles for "Review" in table 5 is 0. Accordingly, probability distribution of activity "Verify" in table 6 is also 0.

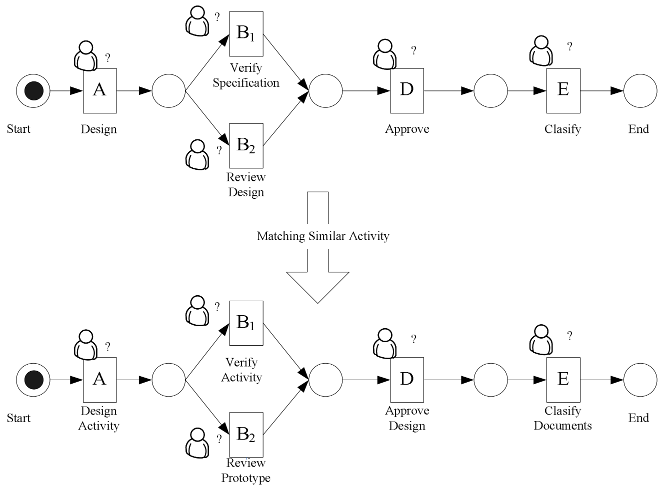

Fig. 4. Target workflow model after matching similar activity

Table 5. Probability distribution of roles over activities with s_1

	Start	*Design Activity*	*Verify Activity*	*Review Prototype*	*Approve Design*	*Classify Docu-ments*	*End*
Start Role	1.000	0.000	0.000	0.000	0.000	0.000	0.000
Senior Designer	0.000	0.9438	0.0113	0.000	0.0007	0.0014	0.000
Chief Designer	0.000	0.0562	0.9572	0.000	0.0002	0.0008	0.000
Project Manager	0.000	0.000	0.0315	0.000	0.9990	0.000	0.000
Secretary	0.000	0.000	0.000	0.000	0.000	0.9978	0.000
end role	0.000	0.000	0.000	0.000	0.000	0.000	1.000

Table 6. Probability distribution of roles over activities with s_2

	Start	*Design Activity*	*Verify Activity*	*Review Prototype*	*Approve Design*	*Classify Documents*	*End*
Start Role	1.000	0.000	0.000	0.000	0.000	0.000	0.000
Senior Designer	0.000	0.9525	0.000	0.0852	0.0024	0.0025	0.000
Chief Designer	0.000	0.0475	0.000	0.9000	0.0005	0.0016	0.000
Project Manager	0.000	0.000	0.000	0.0148	0.9972	0.000	0.000
Secretary	0.000	0.000	0.000	0.000	0.000	0.9959	0.000
end role	0.000	0.000	0.000	0.000	0.000	0.000	1.000

Finally, the probability distribution over roles for all activities can be computed by calculating the normalized weighted average of table 5 and table 6. The result is shown in table 7. With probability distribution shown in table 7, the most suitable role assignment for activities in the new workflow model is shown in figure 5.

Table 7. Probability distribution of roles over all Activities

	Start	*Design*	*Verify*	*Review*	*Approve*	*Classify*	*End*
Start Role	1.000	0.000	0.000	0.000	0.000	0.000	0.000
Senior Designer	0.000	0.9482	0.0113	0.0852	0.0015	0.0019	0.000
Chief Designer	0.000	0.0518	0.9572	0.9000	0.0003	0.0012	0.000
Project Manager	0.000	0.000	0.0315	0.0148	0.9981	0.000	0.000
Secretary	0.000	0.000	0.000	0.000	0.000	0.9969	0.000
end role	0.000	0.000	0.000	0.000	0.000	0.000	1.000

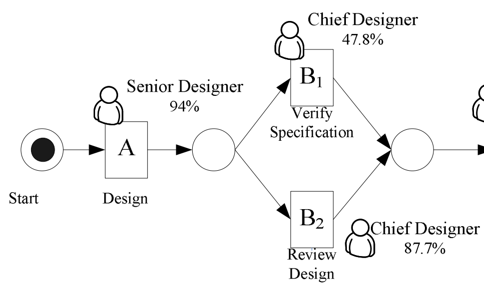

Fig. 5. Suggested role assignment

5 Related Work

The purpose of our work is to automate the part of resource assignment. In particular, we used a probabilistic approach to realize our idea. Therefore it is related to the efforts of automating process resource assignment. Automating resource assignment is very important in workflow resource management [7]. Early work on automating resource assignment in process management is based upon rules [8, 9, 10]. However, rule based approaches of automating resource assignment require knowledge of organization and business, which is not likely to be obtained in the first beginning. In [11] zur Muehlen envisioned the idea of applying knowledge discovery approach to help process resource assignment, later on, in [12] Ly et al. have shown that the problem of deriving resource assignment rules using information from event log data and organizational information as input can be interpreted as an inductive learning problem. Therefore, machine learning techniques can be adapted in order to solve the problem. In particular they use decision tree methods to find those assignment rules [13]. In [14] Liu et al. further developed the approach using new machine learning approaches and evaluated the practical validity using three enterprises' data set. In [15] Huang et al. proposed a reinforcement learning based approach to allocate resource to workflow with performance optimization consideration. They introduce a mechanism in which the resource allocation optimization problem is modeled as Markov decision processes and solved using reinforcement learning. The proposed mechanism observes its environment to learn appropriate policies, which optimize resource allocation in business process execution. The hidden Markov model based approach is also used in [6] by Yang et al. to allocate the most proficient set of employees for a whole business process based on workflow event logs.

6 Conclusion

The assignment of roles to process activities is a time-consuming task and requires a certain amount of business process model experiences. In this paper we have first introduced a meta-model for the description of resources that can be utilized in business processes. The advantage of this model is an exhaustive consideration of roles' skills and competences, thus allowing to allocate appropriate resources (persons that fulfill specific roles) to given activities of workflow models. Based upon this meta-model we used the hidden Markov model inference to provide recommendation for the assignment of roles to process activities. Assisting process modelers to efficiently assign roles to activities is of great value.

Work that is in progress is to integrate a formalism that allows checking role conflicts (if roles are assigned to activities, which are not able to perform the task). Furthermore the consideration of actual resource capacities (number of resources attached to roles) and instance properties (instantiation of workflow instances based on probability distributions) would be valuable and is also part of current research activities.

References

1. Liu, Y., Wang, J., Sun, J.: A Machine Learning Approach to Semi-Automating Workflow Staff Assignment. In: The 22nd Annual ACM Symposium on Applied Computing (SAC 2007), Seoul, Korea (2007)
2. Liu, Y., Wang, J., Sun, J.: Mining Workflow Event Log to Find Parallel Task Dispatching Rules. In: The 4th International Workshop on Computer Supported Activity Coordination (CSAC 2007), Funchal, Madeira - Portugal (2007)
3. Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE 77(2), 257–286 (1989)
4. Cappé, O., Moulines, E., Rydén, T.: Inference in Hidden Markov Models. Springer, Heidelberg (2005)
5. Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: Biological Sequence Analysis (1998)
6. Yang, H., Wang, C., Liu, Y., Wang, J.: An Optimal Approach for Workflow Staff Assignment Based on Hidden Markov Models. In: Meersman, R., Herrero, P. (eds.) OTM-WS 2008. LNCS, vol. 5333, pp. 24–26. Springer, Heidelberg (2008)
7. Russell, N., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Workflow Resource Patterns. Eindhoven University of Technology, Eindhoven (2005)

8. Du, W., Shan, M.-C.: Enterprise Workflow Resource Management. In: HP Labs Technical Reports (1999)
9. Huang, Y.-N., Shan, M.-C.: Policies in a Resource Manager of Workflow Systems: Modeling, Enforcement and Management. In: HP Labs Technical Reports (1999)
10. Reis, C.A.L., Reis, R.Q., Schlebbe, H., Nunes, D.J.: A Policy-Based Resource Instantiation Mechanism to Automate Software Process Management. In: Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering. ACM Press, Ischia (2002)
11. zur Muehlen, M.: Organizational Management in Workflow Applications – Issues and Perspectives. Information Technology and Management 5(3-4), 271–291 (2004)
12. Ly, L.T., Rinderle, S., Dadam, P., Reichert, M.: Mining Staff Assignment Rules from Event-Based Data. In: Bussler, C.J., Haller, A. (eds.) BPM 2005. LNCS, vol. 3812, pp. 177–190. Springer, Heidelberg (2006)
13. Rinderle-Ma, S., van der Aalst, W.: Life-Cycle Support for Staff Assignment Rules in Process-Aware Information Systems, Technical Report, TU Eindhoven (2007)
14. Liu, Y., Wang, J., Yang, Y., Sun, J.: A Semi-Automatic Approach for Workflow Staff Assignment. Comput. Ind. 59(5), 463–476 (2008)
15. Huang, Z., van der Aalst, W., Lu, X., Duan, H.: Reinforcement Learning Based Resource Allocation in Business Process Management. Data & Knowledge Engineering (2010)
16. van der Aalst, W.M.P., Kumar, A., Verbeek, H.M.W.: Organizational modeling in UML and XML in the context of workflow systems. In: Proceedings of the 2003 ACM Symposium on Applied Computing, Melbourne, Florida, pp. S. 603–S. 608 (2003)
17. European e-Competence Framework, "European e-Competence Framework 2.0 – A commom European Framework for ICT Professionals in all industrie sectors". In: CEN Workshop Agreement, European Commission (2010)
18. HR-XML Consortium: Competencies, HR-XML Version 3.0 (2011), `http://www.hr-xml.org/`
19. OASIS: Web Services – Human Task (WS-HumanTask) Specification Version (November 2009)
20. Object Management Group: Meta Object Facility (MOF) Core Specification OMG Available Specification Version 2.0 (January 2006), `http://www.omg.org/docs/formal/06-01-01.pdf`
21. Oberweis, Schuster, T.: A meta-model based approach to the description of resources and skills. In: 16th Americas Conference on Informaction Systems, AMCIS 2010 Proceedings (2010)
22. Ehrig, M., Koschmider, A., Oberweis, A.: Measuring Similarity between Semantic Business Process Models. In: Proceedings of the Fourth Asia-Pacific Conference on Conceptual Modelling (APCCM 2007), pp. 71–80. Australian Computer Science Communications, Ballarat (2007)